International Journal of
Operations Research

International Journal of Operations Research Vol. 9, No. 1, 53−65 (2012)

A Hybrid Electromagnetism-like Mechanism: A Metaheuristic Algorithm for Solving the Travelling Salesman Problem

Peitsang Wu[1,*], Yung-Yao Hung[1] and, Han-Chieh Chiang[1]

[1] Department of Industrial Engineering and Management, I-Shou University,

No.1, Sec. 1, Syuecheng Rd., Dashu District, Kaohsiung City 84001, Taiwan, R.O.C.

Received April 2011; *Revised February 2012*; *Accepted March 2012*

***Abstract*—** The Electromagnetism-like Mechanism (EM) is a metaheuristic algorithm which utilizes an attraction-repulsion mechanism to move the sample points (i.e., our solutions) towards optimality. Birbil *et al.* (2005) have verified that the EM algorithm can avoid the solutions leading to the local minimum and move toward the global optimum. This study was undertaken to ascertain the effects of using the proposed hybrid EM algorithm in regard to its ability to solve the travelling salesman problem (TSP). In this paper, the authors present a hybridization of the EM algorithm and intensive methods which includes the Opt method and the 2-opt method to solve TSPs. Furthermore, since the original EM algorithm is designed to solve real-value-solution problems, this paper modifies our hybrid EM algorithm with a Random-Key technique for solving the TSP which, specifically, is an integer-valued-solution problem. The computational results show that the proposed hybrid EM algorithm is capable of solving the TSP.

***Keywords*—** Electromagnetism-like mechanism (EM), travelling salesman problems (TSP).

1. INTRODUCTION

Recently, there has been a dramatic increase in the number of publications on combinatorial optimization problems. One of the most common problems is the travelling salesman problem (TSP), which is similar to NP-Complete problems. Recently, many metaheuristic algorithms have been employed to solve TSPs. The metaheuristic algorithms include: simulated annealing (Kirkpatrick *et al.*, 1985; Lo and Hus, 1998; Tian and Wang, 2000; Meer, 2007), Tabu search (Knox, 1989; Glover, 1990; Knox,1994), genetic algorithms (Potvin, 1996; Jiao and Wang, 2000; Baraglia *et al.*, 2001; Moon *et al.*, 2002; Yang *et al.*,2008), scatter search (Liu, 2007; Liu, 2008), particle swarm optimization (Shi *et al.*, 2007; Marinakis and Marinaki, 2010; Marinakis *et al.*, 2010), and ant colony optimization (Tsai *et al.*, 2004; Cheng and Mao, 2007; López-Ibáñez and Blum, 2010; Ghafurian and Javadian, 2011). The resulting solutions from these metaheuristic algorithms are capable of finding solutions near the optimum and even the optimal solutions in some special cases. As such, the solutions from the metaheuristic algorithms may not always be optimum but they are cost-effective in terms of time and computer processing and memory loads.

The Electromagnetism-like Mechanism (EM) algorithm is a metaheuristic algorithm proposed and developed by Birbil and Fang (2003). The EM algorithm simulates the attraction-repulsion mechanism in electromagnetism theory. The EM algorithm has been tested and verified; it can converge rapidly (in terms of the number of function evaluations) on the global optimum and produce highly efficient results in regard to problems showing varying degrees of difficulty (Birbil and Fang, 2003; Birbil *et al.*, 2005).

Also, the EM algorithm was applied to NP-Hard problems such as: scheduling problems (Debels and Vanhoucke, 2006; Maenhout and Vanhoucke, 2007; Chang *et al.*, 2009; Naderi *et al.*, 2010; Jamili *et al.*, 2011), TSP problems (Javadian *et al.*, 2008), VRP problems (Yurtkuran and Emel, 2010), etc, where the results were promising. However, many metaheuristic algorithms have been proposed to solve TSPs, but little attention has been given to the EM algorithm for solving them. Although Javadian *et al.* (2008) proposed a discrete binary version of the EM algorithm for solving small size TSPs having a smaller number of iterations, medium to large size TSPs have not been tested. Thus, we were motivated to apply the EM algorithm to solve a wider variety of travelling salesman problems.

This paper presents a hybridization of the EM algorithm and intensive methods which include the Opt method and the 2-opt method in the context of solving TSPs. Furthermore, since the original EM algorithm is designed to solve real-value-solution problems, this paper proposes to modify our hybrid EM algorithm with a Random-Key (RK) technique to solve the TSP, which is an integer-valued-solution problem.

[*] Corresponding author's email: pwu@isu.edu.tw

The rest of this paper is organized as follows. Section 2 describes the TSP and lays out its mathematical formulation. The hybrid EM algorithm with the RK technique and intensive K-opt methods are proposed in Section 3. Computational results are given in Section 4. We offer conclusions in Section 5.

2. TRAVELLING SALESMAN PROBLEMS

The TSP is a classical combinatorial optimization problem, which is simple but very difficult to solve. The simplest TSP involves finding an optimal tour, i.e., the shortest tour wherein a salesman is able to visit *n* cities and finally return to the point of departure. Furthermore, each city is visited exactly once and the inter-city distances are symmetric and known. Here, we discuss the simplest TSP by converting the graph into a mathematical formulation. The constituents of the TSP are such that: *n* is the number of cities indexed by *i* and *j*, $i, j \in \{1, \ldots, n\}$; c_{ij} is the distance between city *i* and *j*; x_{ij} is the decision variables. The x_{ij} equal to 1 when arc (*i, j*) is included in the tour, and equal to 0 otherwise. The TSP can be represented as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1, \ldots, n. \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1, \ldots, n. \tag{3}$$

$$(x_{ij}) \in X, \tag{4}$$

$$x_{ij} = 0 \text{ or } 1, \qquad \forall i,\ j = 1, \ldots, n. \tag{5}$$

The objective function (1) represents the total tour for the travelling salesman. Constraints (2) and (3) ensure that each city is visited exactly once. Constraint (4) denotes the solution where we found that (x_{ij}) was located in the feasible region X. Constraint (5) denotes that the x_{ij} are the binary numbers of the integer.

3. HYBRID ELECTROMAGNETISM-LIKE MECHANISM METAHEURISTIC ALGORITHM FOR USE WITH TRAVELLING SALESMAN PROBLEMS

Birbil and Fang (2003) constructed the EM that is compatible with the attraction-repulsion mechanism of the electromagnetism theory and as such, is able to process each sample point (solution) as it is released into a space as a charged particle whose charge relates to the objective function value. The charge determines the magnitude of attraction or repulsion of the point over the sample population, i.e., the better the objective function value, the higher the magnitude of attraction. After calculating these charges, we can find a direction that is derived from a combination force, such as electromagnetic forces, which is calculated by adding, in a vector-wise manner, the forces from each of the other points calculated separately. The attraction directs the points towards better regions, whereas repulsion allows particles to exploit the unvisited regions.

3.1 General Scheme

We apply the EM to the following global optimization problems with bounded variables:

$$\min f(x) \qquad s.t.\ x \in \left[l, u\right] \ , \tag{6}$$

where $\left[l,u\right] := \left\{ x \in \Re^n \ \middle|\ l_k \le x_k \le u_k \ ,\ k = 1, \ldots n \right\}$, and the parameters are defined as:

n: dimension of the problem (the number of cities)
u_k: upper bound in the k^{th} dimension
l_k: lower bound in the k^{th} dimension
$f(x)$: pointer to the function which is minimized.

Table 1. General scheme

ALGORITHM EM (m, $MAXITER$, $LSITER$, δ)
m : number of sample points
MAXITER: maximum number of iterations
LSITER: maximum number of local search iterations
δ : local search parameter, $\delta \in [0,1]$
1. Initialize( )
2. iteration ← 1
3. **while** (iteration< $MAXITER$) **do**
4. Local($LSITER$, δ)
5. **F**←CalcF()
6. Move(**F**)
7. iteration ← iteration +1
8. end **while**

3.1.1 Initialize

Table 1 illustrates the General Scheme of the EM algorithm. The procedure *initialize* generates m sample points (solutions) randomly from the feasible domain, which is an n dimensional hyper-cube. Each coordinate of a point is assumed to be uniformly distributed between the corresponding upper bound and lower bound. After a point is sampled from the space, the objective function value for the point is calculated using the function pointer $f(x)$. The procedure ends with m points identified, and the point that has the best function value is stored in x^{best}. Notably, the words *particle* and *point* are interchangeably used.

3.1.2 Local search

The procedure *local search* is used to gather the local information for a point x^i. The procedure is as follows: first, a length is calculated by the maximum difference of each dimension's upper and lower bound. Here, the procedure makes use of the parameter $\delta \in [0, 1]$ to derive a feasible random length (*length*). Second, a temporary point y is used to store the initial point x^i and let the point y move along the direction according to the feasible random length coordinate by coordinate. Next, if the point y observes a solution within the *LSITER* iterations, the point x^i is replaced by y and the neighborhood search for point x^i ends. Finally, the x^{best} is updated.

3.1.3 Total force calculation

Cowan (1968) suggested that the force exerted on a point via other points is inversely proportional to the distance between the points and directly proportional to the product of their charges; the total force on each particle is calculated in the spirit of Coulomb's Law. The charge of each point, q^i, determines point i's power of attraction or repulsion. This charge is evaluated by:

$$q^i = \exp\left(-n\frac{f(x^i)-f(x^{best})}{\sum_{k=1}^{m}\left[f(x^k)-f(x^{best})\right]}\right),\ i=1,2,...m. \tag{7}$$

where $f(x^i)$ is the objective function value of each point x^i and the current best point x^{best} in the population has better objective function values $f(x^{best})$ resulting in higher charges. The parameter n is the dimension of the solution space. Notice that, unlike electrical charges, no signs are attached to the charge of an individual point in Equation (7).

After calculating the charge of each point or solution, the total force F^i exerted on point x^i is computed by the following:

$$F^i = \sum_{\substack{j=1\\ j\neq i}}^{m}\left\{\begin{array}{ll}(x^j-x^i)\cdot\dfrac{q^iq^j}{\left\|x^j-x^i\right\|^2} & \text{if } f(x^j)<f(x^i)\\ (x^i-x^j)\cdot\dfrac{q^iq^j}{\left\|x^j-x^i\right\|^2} & \text{if } f(x^j)\geq f(x^i)\end{array}\right\},\ i=1,2,...m. \tag{8}$$

3.1.4 Move along the total force

After evaluating the total force F^i, the point is designed to move in the direction of the force by a random step length in Equation (9). Here, two parameters must be defined: one is the random step length, λ, and is assumed to be uniformly distributed between 0 and one; the other is the *RNG* whose components denote the allowed feasible movement toward the upper bound u_k, or the lower bound, l_k, for the corresponding dimension. According to the above mechanism, EM ensures that points have a nonzero probability with regard to the unvisited regions along this direction.

$$x^i = x^i + \lambda \frac{F^i}{\left\| F^i \right\|}(RNG) \qquad i = 1,2,\ldots,m. \tag{9}$$

where RNG denotes the allowed range of movement toward the lower bound l_k, or the upper bound u_k, for the corresponding dimension. We adopt the notation x_k^i to specify the k^{th} dimension of the i^{th} point.

$$RNG = \begin{cases} u_k - x_k^i & if\ F_k^i > 0 \\ x_k^i - l_k & if\ F_k^i < 0 \end{cases} \qquad k = 1,2,\ldots,n. \tag{10}$$

3.2 The Modified EM with a Random-Key for the TSP

As we know, the EM algorithm is designed for real-value-solution problems. In order to find the integer-valued solution of the TSP, we use the Random-Key (RK) technique. In RK representation, the k^{th} dimension's value is a priority value for the k^{th} activity. Figure 1 demonstrates a 10-dimension solution. Values of dimension 1 to 10 are shown in Figure 1. Then, we apply the RK technique to sort these values in ascending order. Thus, the sequence at position 1 is 10 means we can schedule City 10 in the beginning and City 3 in the last position. Notably, through the RK technique, the modified EM algorithm is capable of solving different kinds of sequencing problems.

Here, we perform a tour of the TSP representation via RK form. Through this process, we can observe Constraints (2) and (3). Because the priority of the city is taken according to the value which it is given, the method gives an index such that a lower value has a higher priority in the tour. In RK form, a solution corresponds to a point in the Euclidian space where each dimension is a parameter's value, similar to the representation which EM uses. Consequently, we change the value of each dimension according to the electromagnetic force which corresponds to the objective function value. The modified EM for the TSP will be detailed below. In *Initialization*, because we want to decrease the solution space of the search space, we set the $l_k=1$ and $u_k=n$ (n means we have n cities) and randomly produce priority values for each dimension. Through this procedure, we can obtain the feasible priority structure of a tour. In *Total force calculation*, we determine the charge q^i and total force F^i of the solution x^i according to Equations (7) and (8). We change the priority base on the value after using the *Move along the total force* method. Table 2 displays a tour of RK form found after applying the serial methods of our modified EM algorithm.

Activities

(Position)	1	2	3	4	5	6	7	8	9	10
Before	8.28	6.34	2.30	1.35	4.11	4.03	5.52	5.11	4.02	3.12

(a) Value of activities

After	10	9	2	1	6	5	8	7	4	3

(b) Schedule list

Figure 1. A demonstration of the RK technique with a 10-dimension solution

Table 2. Illustration an RK form of the modified EM algorithm

x_k^i	8.28	6.34	2.30	1.35	4.11	4.03	5.52	5.11	4.02	3.12
Initial Tour (City)	10	9	2	1	6	5	8	7	4	3
$\frac{F_k^i}{\lVert F_k^i \rVert}$	0.03	0.04	-0.04	-0.07	-0.05	-0.01	0.04	0.01	0.01	-0.01
$\Delta x_k^i = \lambda \frac{F_k^i}{\lVert F_k^i \rVert}(RNG)$	0.28	0.88	-0.30	-0.15	-0.92	-0.23	0.98	0.36	0.41	-0.09
$x_k^i \leftarrow x_k^i + \Delta x_k^i$	8.56	7.22	1.99	1.21	3.20	3.80	6.50	5.48	4.43	3.04
New Tour (City)	10	9	2	1	4	5	8	7	6	3

3.3 Hybridization of Modified EM algorithm and Intensive Methods

In this section, we introduce two intensive methods which are the *Opt* method and the *2-opt* method. These methods are similar to the tour improvement procedures and significantly improve the efficiency of the algorithm. A detailed discussion will be presented later.

3.3.1 The Opt Method

The Opt method is one of the tour improvement procedures which is used. Two main mechanisms, namely, 2-opt and 4-opt are integrated into the Opt method. Figures 2 and 3 each show examples of 2-opt and 4-opt exchange techniques, respectively. We use the Opt method to find a new tour even it is not good, and provide an important parameter, *Ls*, for the EM algorithm. This parameter, *Ls*, is the number of the search iterations for each point, namely, all solutions. We also set a probability to choose the 2-opt or 4-opt procedure randomly in the Opt method. We implement the Opt method in Table 3, which is the main procedure of the Opt method. Note, the Opt method does not follow all the steps of the original 2-opt and 4-opt heuristics because we also leave room to accept the solutions which are not good except for the best solution.

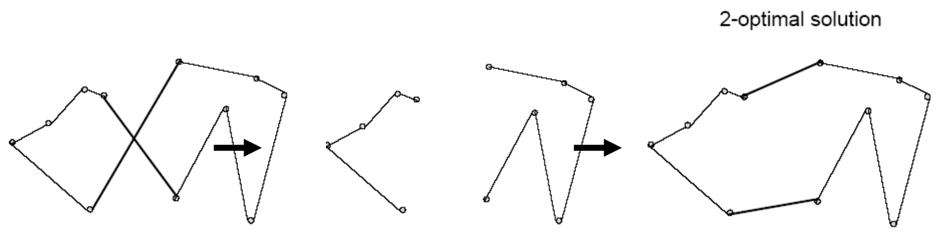

Figure 2. The 2-opt method

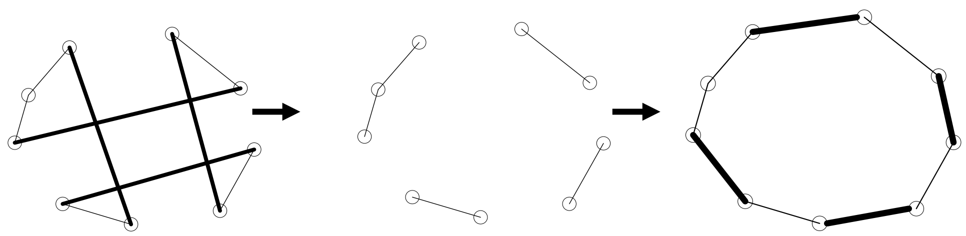

Figure 3. The 4-opt method

Table 3. Main procedure of the Opt method

Opt Method (Ls, δ)

Ls : number of search iterations for each point
δ_1 : Opt method parameter, $\in [0,1]$
δ_2 : Opt method parameter, $\in [0,1]$

$iter \leftarrow 1$
for i=1 to m **do**
  while $iter < Ls$ **do**
    $\delta_1, \delta_2 \leftarrow U(0,1)$
    if $\delta_1 > 0.5$ **then**
      $f(y) \leftarrow$ 2-opt()
      if $f(y) < f(x^i)$ **then**
        $x^i \leftarrow y$
      else
        if ($i \neq best$) and ($\delta_1 < \delta_2$) **then**
          $x^i \leftarrow y$
    else
      $f(y) \leftarrow$ 4-opt()
      if $f(y) < f(x^i)$ **then**
        $x^i \leftarrow y$
      else
        if ($i \neq best$) and ($\delta_1 > \delta_2$) **then**
          $x^i \leftarrow y$
    $iter \leftarrow iter + 1$
  end while
end for
$x^{best} \leftarrow \arg\ \min\left\{f(x^i, \forall i)\right\}$

3.3.2 The 2-opt Method

The 2-opt method is another tour improvement procedure which is used in this study. The main mechanism of this method is the 2-opt improvement heuristic. The exchange method of the 2-opt heuristic has been performed in Figure 2 and we follow the steps of the prime 2-opt heuristic to improve the tour. Table 4 shows the procedure of the 2-opt method, i.e. the 2-opt heuristic.

Table 4: Main procedure of the 2-opt method

2-opt Method ()

for i=1 to m **do**
  for j=1 to n **do**
    $f(y) \leftarrow$ 2-opt()
    if $f(y) < f(x^i)$ **then**
      $x^i \leftarrow y$
  end for
end for
$x^{best} \leftarrow \arg\ \min\left\{f(x^i, \forall i)\right\}$

4. ILLUSTRATED EXAMPLES & ANALYSES

Here, we use the EM algorithm such that the EM algorithm individually utilizes the Opt and the 2-opt methods to solve the 16-cities problem. We set up the important parameters: *m*=10, *MAXITER*=1000, *LSITER*=100 and Ls=100. Note that there are two variations of the Opt methods, namely, the EM+Opt (1) and the EM+Opt (2). There is a difference between these two methods. In the EM+Opt (1), the local search is used in each iteration but in the EM+Opt (2) it is not, i.e., the Opt method replaces the local search. The fourth and fifth methods are such that the basic EM utilizes the 2-opt method. There are also two variations of the 2-opt methods: the EM+2-opt (1) and the EM+2-opt (2). There is also a difference between these two methods. In EM+2-opt (1), the local search is used in each iteration but in the EM+2-opt (2) it is not, i.e., the 2-opt method is substituted for the local search.

Table 5. Results of five different methods

Method	Best result **(a)**	Average (20 trials)	Optimal result **(b)**	% Error **((a)-(b)) / (b)**	Avg. CPU (seconds)
Basic EM	3.6129	4.0114	3.2 (0/20)	12.90%	0.6815
EM+Opt (1)	3.2	3.3449	3.2 (8/20)	0%	5.1689
EM+Opt (2)	3.2	3.3861	3.2 (5/20)	0%	5.1323
EM+2-opt (1)	3.2	3.2414	3.2 (15/20)	0%	11.3245
EM+2-opt (2)	3.2	3.2497	3.2 (14/20)	0%	11.5972

After these five methods were used to solve the 16-cities problem, the results are presented in Table 5. From this table, the best results are presented; also the optimum is 3.2 for each method except for the *Basic EM*. Therefore, the *% Error* corresponding to each revised method is also 0%. In Figure 4, we can see that the convergence lines of the four revised methods fell rapidly until they reached 3.2, while the convergence lines of the *Basic EM* did not. Here, there are two phenomena which we should notice. The first phenomenon is that the value of *Average* is close to 3.2 when the number of optimal solutions successfully found is increased. Although the trials are not 100% optimum, we obtain much better results in the 16-cities problem while incorporating the Opt and 2-opt methods. The second phenomenon is such that we can obtain the optimal results by using the EM+Opt and the EM+2-opt methods, but more time must be spent adding the tour improvement procedures. The main reason for this phenomenon is that the Opt or 2-opt methods are iterated for all sample points or solutions.

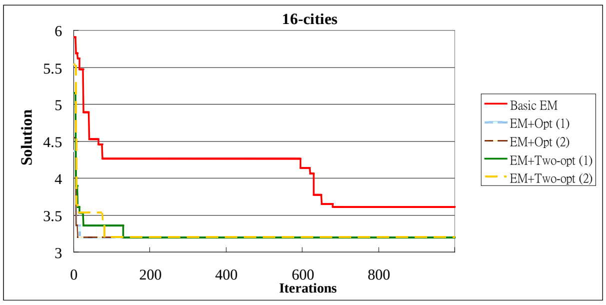

Figure 4. Convergence diagram (five different methods)

4.1 Design and Analysis of Experiments for 16 Cities

In this section, we discuss how the parameters in the EM procedure impact the performances of the *Basic EM*, the EM+Opt, and the EM+2-opt methods for the TSPs. Therefore, the 3^3 factorial designs are analyzed for the three responses of the TSPs: the *Average*, *% Error,* and *Avg. CPU*, respectively. First, we select three important parameters in the EM procedure of our experiment, each at three levels. The first parameter is *m*, the number of sample points. The second parameter is *MAXITER*, maximum number of iterations. The last parameter involves the methods which we use to solve the TSP. The three levels of the parameters (numerical or text) are defined below (see Table 6). Finally, we can obtain the results of *Average*, *% Error* and *Avg. CPU* for the 16-cities problem (see Tables 7-9).

Table 6. The levels of the parameters

Sample points (m)	Iterations (*MAXITER*)	Methods
5	500	EM
10	1000	EM+Opt
20	2000	EM+2-opt

Table 7. Results of the Average

		Methods		
Sample points	Iterations	EM	EM+Opt	EM+2-opt
5	500	4.4571	3.2994	3.3820
	1000	4.0226	3.4320	3.2826
	2000	4.0460	3.3157	3.3326
10	500	4.1537	3.2826	3.2331
	1000	3.9219	3.4320	3.3157
	2000	3.9664	3.4483	3.2331
20	500	3.5799	3.3489	3.2
	1000	3.6837	3.3657	3.2
	2000	3.4951	3.3326	3.2331

Table 8. Results of the % error

		Methods		
Sample points	Iterations	EM	EM+Opt	EM+2-opt
5	500	0.2063	0	0.0518
	1000	0.2063	0.0518	0
	2000	0.1250	0	0
10	500	0.1402	0	0
	1000	0.1036	0	0
	2000	0.1656	0.0518	0
20	500	0.0773	0	0
	1000	0.1036	0	0
	2000	0.0518	0	0

Table 9. Results of the avg. CPU

		Methods		
Sample points	Iterations	EM	EM+Opt	EM+2-opt
5	500	0.1250	1.4058	3.3246
	1000	0.2402	2.8118	6.4840
	2000	0.4684	5.6528	13.2870
10	500	0.3590	2.9996	6.6340
	1000	0.6868	5.9746	13.2338
	2000	1.3842	12.1120	26.7278
20	500	1.2246	6.4526	13.5184
	1000	2.3966	12.7902	27.0120
	2000	4.8154	25.7434	54.3338

We used the statistical software, MINITAB, to design our experiments. The confidence coefficient is 0.95 (i.e., α=0.05). According to the data of these responses, we may consider the ANOVA for the responses under the following tables: (1) the ANOVA for *Average*; (2) the ANOVA for *% Error*; (3) the ANOVA for *Avg. CPU*.

Table 10 is an analysis of variance for the *Average*. We find that the P value (=0.005) of the *Sample points*Methods* is less than 0.05. That is, we have the interaction between the sample points and methods. Since the interaction *Sample points*Methods* is significant, we plot an interaction for it (see Figure 5). From the figure, we have the following observation. When using the EM method to solve the 16-cities problem, the sample points equal to 20 have a much lower objective value with regard to the

Average and show a significant difference as compared to those where the sample points are equal to 5 or 10. This circumstance is not significant when we use the EM+Opt or the EM+2-opt methods.

Table 11 is an analysis of variance for the *% Error*. We find that the P values of *Sample points* (=0.01) and *Methods* (=0.00) are less than 0.05, which means these two factors have main effects for the *% Error*. Consequently, we have the main effects plot of these two factors in Figure 6. From this figure we can clearly understand these two factors' effects upon the *% Error*. When the sample point is set to 20 or the method is set to the EM+2-opt, we can obtain the smaller value of the *% Error* than those levels of other variables.

Table 10. ANOVA for the Average

Source	DF	SS	MS	F	P
Sample points	2	0.2695	0.1348	15.0900	0.0020
Iterations	2	0.0158	0.0079	0.8900	0.4490
Methods	2	2.2738	1.1369	127.3300	0.0000
Sample points*Iterations	4	0.0304	0.0076	0.8500	0.5320
Sample points*Methods	4	0.3117	0.0779	8.7300	**0.0050**
Iterations*Methods	4	0.0878	0.0220	2.4600	0.1300
Error	8	0.0714	0.0089		
Total	26	3.0603			

Table 11. ANOVA for the % error

Source	DF	SS	MS	F	P
Sample points	2	0.0093	0.0047	8.5500	**0.0100**
Iterations	2	0.0004	0.0002	0.4000	0.6830
Methods	2	0.0901	0.0450	82.6500	**0.0000**
Sample points*Iterations	4	0.0062	0.0015	2.8400	0.0970
Sample points*Methods	4	0.0075	0.0019	3.4400	0.0640
Iterations*Methods	4	0.0021	0.0005	0.9500	0.4850
Error	8	0.0044	0.0005		
Total	26	0.1200			

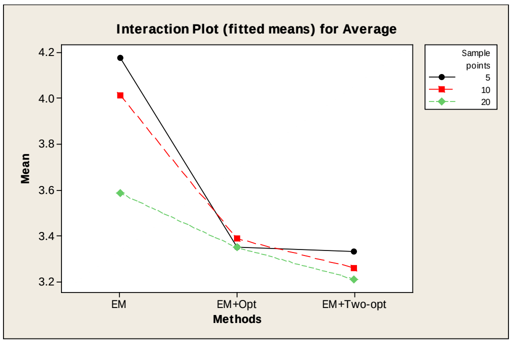

Figure 5. Interactions between the sample points and the methods

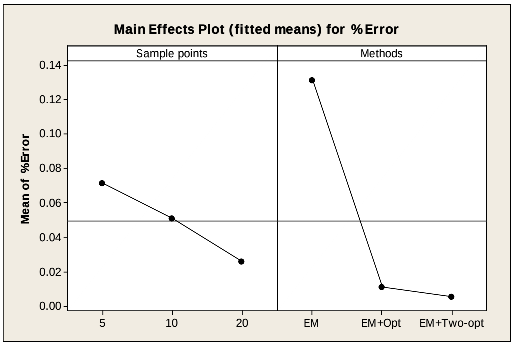

Figure 6. Main effects of the sample points and the method

Finally, Table 12 is an analysis of variance for the *Avg. CPU*. We find that the P values of *Sample points*Iterations*, *Sample points*Methods*, and *Iterations*Methods* are less than 0.05. That means we have interaction between these factors. Figure 7 shows the interaction plots for the *Sample points*Iterations*, *Sample points*Methods*, and *Iterations*Methods*. Figure 7 (a) is the interaction plot for the *Sample points*Iterations*, Figure 7 (b) is the interaction plot for the *Sample points*Methods* and Figure 7 (c) is the interaction plot for the *Iterations*Methods*. In Figure 7 (b), we can see that the sample points equal to 20 have a much higher value of *Avg. CPU* and have significant differences when compared with sample points are equal to 5 or 10 in the EM or the EM+Opt methods and in the iterations equal to 1000 or 2000. In Figure 7 (c), the iterations equal to 2000 have much higher value of *Avg. CPU* and have significant differences when compared with iterations equal to 500 or 1000 in the EM+Opt, or the EM+2-opt methods.

Table 12. ANOVA for the avg. CPU

Source	DF	SS	MS	F	P
Sample points	2	760.6400	380.3200	29.6300	0.0000
Iterations	2	679.5600	339.7800	26.4700	0.0000
Methods	2	1309.0400	654.5200	50.9900	0.0000
Sample points*Iterations	4	217.9500	54.4900	4.2400	**0.0390**
Sample points*Methods	4	357.0700	89.2700	6.9500	**0.0100**
Iterations*Methods	4	379.4700	94.8700	7.3900	**0.0090**
Error	8	102.6900	12.8400		
Total	26	3806.4300			

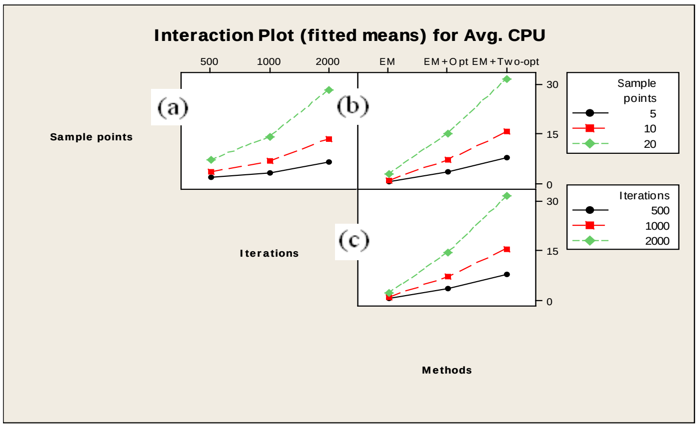

Figure 7. Interactions for the "avg. CPU"

4.2 Illustrative Examples with 30, 50, 75 and 100 Cities

We have solved the 16-cities problem and obtained the optimal solutions excluding that of the original EM algorithm. The performances of the EM+Opt and the EM+2-opt methods for the 16-cities problems are satisfactory. Therefore, we are interested in using these methods to solve large-city problems. The performance measures of such methods are compared with the performance measures of other naturally inspired global optimization methods, such as genetic algorithms (GAs), ant colony systems (ACSs), and simulated annealing (SA). Here, we use the EM+Opt or EM+2-opt methods to solve the Oliver30 (30-cities problem), Eil50 (50-cities problem), Eil75 (75-cities problem) and KroA100 (100-cities problem), which are from Dorigo and Gambardella (1997). According to the ANOVA for the data contained in the previous tables (Tables 10 to 12), we set up such that *Sample points* = 20, *Iterations* = 500, and *Methods* = EM+Opt.

Table 13 reports the results of the EM+Opt method which we use to solve the problems. We report the best tour obtained from 20 trials in *Best result* and the best real tour length (Dorigo and Gambardella, 1997) in parentheses. The deviation from the optimal result and the average computation time are also shown in the *% Error* and the *Avg. CPU* columns, respectively. First, we can find that the larger a city we want to solve for, the more time the method will take because that the EM algorithm calculates the total force, F^i, and moves coordinate by coordinate. Second, the larger problem we want to solve, the more combinatorial solutions we have. Therefore, we find that the values of *% Error* increase when we want to solve a large-city problem. Finally, the performance of the EM+Opt method for solving the large-city problems is not as good as for solving small-city problems, but we still can find the optimal solution, i.e., the shortest tour, of the Oliver30 problem.

Table 13. Results of the EM+Opt method for the large-city problems

Problem name	EM+Opt(1)			EM+Opt(2)		
	Best tour	% Error	Avg. CPU (seconds)	Best tour	% Error	Avg. CPU (seconds)
Oliver30 (30-cities problem)	423.949 (423.74)	0.05%	14.6409	423.74 (423.74)	**0.00%**	14.5004
Eil50 (50-cities problem)	442.081 (427.86)	3.32%	38.3731	437.147 (427.86)	2.17%	37.4996
Eil75 (75-cities problem)	565.278 (542.31)	4.24%	83.3598	580.391 (542.31)	7.02%	80.5230
KroA100 (100-cities problem)	22644.8 (21285.44)	6.39%	147.7480	22900.2 (21285.44)	7.59%	141.1427

Here, we also have the comparison of EM with ACS, GA, and SA which are the current state-of-the-art heuristics. We report the best real tour length, the best integer tour length (in parentheses) and the number of tours required to find the best tour length (in square brackets). Because some papers used the coordinate system for calculating the real valued solutions and some papers used the integer distance matrix for calculating the integer valued solutions. Therefore, some of the results of the best tours are enumerated in real values. However, some results of the best tours are enumerated in integer values. The value of "N/A" means the papers didn't provide such information. The performance measures used in the ACS, GA and SA are the results from (Dorigo and Gambardella, 1997). The EM+Opt method was run for 500 iterations with sample points = 20 and had 20 trials. In Table 14, although we do not have the integer results, we can find that the best real tour length (423.74) corresponds to the best integer tour length (420). We can see that the result of EM+Opt (2) for Oliver30 problem reached the optimal outcome. The result is better than those using GA and SA algorithms. In other problems, we can see from the table that the results of the heuristics, except for the ACS, are not as good as the optimum. However, we can gain the numbers of tours required to find the best tour length (in square brackets) of the EM+Opt method. The best tour lengths are less than those of other heuristics. Notably, we compared the results and the number of tours required to find the best tour length with SA and found that we can obtain the better results with a fewer number of tours than SA.

Table 14. Comparison of EM+Opt with the current state-of-the-art heuristics

Problem name	EM+Opt (1)	EM+Opt (2)	ACS	GA	SA	Optimum
Oliver30 (30-cities problem)	423.95 (N/A) **[1600]**	**423.74** (N/A) **[500]**	423.74 (420) [830]	N/A (421) [3200]	N/A (424) [24617]	423.74 (420)
Eil50 (50-cities problem)	442.08 (N/A) **[7500]**	437.15 (N/A) **[1400]**	427.96 (425) [1830]	N/A (428) [25000]	N/A (443) [68512]	N/A (425)
Eil75 (75-cities problem)	565.28 (N/A) **[4100]**	580.39 (N/A) **[8500]**	542.31 (535) [3480]	N/A (545) [80000]	N/A (580) [173250]	N/A (535)
KroA100 (100-cities problem)	22644.80 (N/A) **[9900]**	22900.20 (N/A) **[9200]**	21285.44 (21282) [4820]	N/A (21761) [103000]	N/A (N/A) [N/A]	N/A (21282)

5. CONCLUSIONS

Here, a hybrid EM algorithm for TSPs is introduced and performed. We have illustrated examples which can be solved using the original EM algorithm and two variant methods, the Opt and 2-opt methods, which are comparable with the original EM algorithm. After comparing the results from the computer simulations of these problems, we can conclude that the performance of the EM+Opt and the EM+2-opt methods are good insofar as being able to obtain optimal solutions when trying to solve the 16-cities problem. According to the analysis of variance, we used the EM+Opt method to solve the large-city problems and the performance of this method is shown to be satisfactory for solving the Oliver30 problem. Finally, we provided a comparison of the EM+Opt method with the ant colonies systems (ACS), genetic algorithms (GA), and simulated annealing (SA) which represent the current state-of-the-art heuristics. When solving the Oliver30 problem, the performance of the EM+Opt is better than those using the GA and SA heuristics. In other problems, the numbers of tours required to find the best tour length (in square brackets) for the EM+Opt method are less than those of other heuristics with no worse performance measures, especially as compared with SA. The computational results show that the hybrid EM algorithm is fully capable of solving the TSP problems. Notably, future research could include a hybridization of the EM with other algorithms to achieve even better results.

ACKNOWLEDGEMENT

This research has been supported by the National Science Counsel of R. O. C. under grant NSC 95-2416-H-214-011.

REFERENCES

1. Baraglia, R., Hidalgo, J.I. and Perego, R. (2001). A Hybrid Heuristics for the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 5 (6): 613–622.
2. Birbil, S. I., and Fang, S.-C. (2003). Electromagnetism-like Mechanism for Global Optimization. *Journal of Global Optimization*, 25: 263-282.
3. Birbil, S. I., Fang, S. C., and Sheu, R. L. (2005). Convergence of a Population Based Global Optimization Algorithms. *Journal of Global Optimization*, 30, 301–318.
4. Chang, P. C., Chen S.H., and Fan, C.Y. (2009). A Hybrid Electromagnetism-Like Algorithm for Single Machine Scheduling Problem. *Expert Systems with Applications*, 36:1259–1267.
5. Cheng, C.B. and Mao, C.P. (2007). A Modified Ant Colony System for Solving the Traveling Salesman Problem with Time Windows. *Mathematical and Computer Modeling*, 46: 1225–1235.
6. Cowan, E.W. (1968). *Basic Electromagnetism*, New York: Academic Press.
7. Debels, D. and Vanhoucke, M. (2006). The Electromagnetism Meta-Heuristic Applied to the Resource Constrained Project Scheduling Problem. *Lecture notes in Computer Science*, 3871: 259–270.
8. Dorigo, M. and Gambardella, L.M. (1997). Ant Colonies for the Traveling Salesman Problem. *Biosystems*, 43: 73-81.
9. Ghafurian, S., and Javadian, N. (2011). An Ant Colony Algorithm for Solving Fixed Destination Multi-Depot Multiple Traveling Salesman Problems. *Applied Soft Computing*, 11(1): 1256-1262.
10. Glover, F. (1990). Artificial Intelligence, Heuristics Frameworks and Tabu Search. *Managerial & Decision Economics*, 11: 365-378.
11. Jamili, A., Shafia, M.A., and Tavakkoli-Moghaddam, R. (2011). A Hybridization of Simulated Annealing and Electromagnetism-Like Mechanism for a Periodic Job Shop Scheduling Problem. *Expert Systems with Applications*, 38: 5895-5901
12. Javadian, N., Alikhani, M.G., and Tavakkoli-Moghaddam, R. (2008). A Discrete Binary Version of the Electromagnetism-Like Heuristic for Solving Traveling Salesman Problem. *Lecture Notes in Computer Science*, 5227: 123-130.
13. Jiao, L. and Wang, L. (2000). A Novel Genetic Algorithm Based on Immunity. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, 30 (5):826-830.
14. Kirkpatrick, S., Gelatt Jr. C.D., and Vecchi, M.P. (1985). Configuration Space Analysis of Traveling Salesman Problem. *Journal Physique*, 46: 1277-1292.
15. Knox, J.E. (1989). *The Application of Tabu Search to the Symmetric Traveling Salesman Problem*, Doctoral Dissertation, University of Colorado.
16. Knox, J.E. (1994). The Application of Tabu Search to the Symmetric Traveling Salesman Problem. *Computers and Operations Research*, 21: 867-874.
17. Liu, Y.H. (2007). A Hybrid Scatter Search for the Probabilistic Traveling Salesman Problem. *Computers and Operations Research*, 34(10): 2949-2963.
18. Liu, Y.H. (2008). Diversified Local Search Strategy under Scatter Search Framework for the Probabilistic Traveling Salesman Problem. *European Journal of Operational Research*, 191(2): 332-346.
19. Lo, C.C. and Hus, C.C. (1998). Annealing Framework with Learning Memory. *IEEE Transactions on Systems, Man, and Cybernetics- Part A Systems and Humans*, 28 (5) : 1-13.

20. López-Ibáñez, M. and Blum, C. (2010). Beam-ACO for the Travelling Salesman Problem With Time Windows. *Computers and Operations Research*, 37: 1570-1583.
21. Maenhout, B. and Vanhoucke, M. (2007). An Electromagnetic Meta-Heuristic for The Nurse Scheduling Problem. *Journal of Heuristics*, 13 (4):359-385.
22. Marinakis, Y. and Marinaki, M. (2010). A Hybrid Multi-Swarm Particle Swarm Optimization Algorithm for the Probabilistic Traveling Salesman Problem. *Computers and Operations Research*, 37: 432-442.
23. Marinakis, Y., Marinaki, M., and Dounias, G. (2010). A Hybrid Particle Swarm Optimization Algorithm for the Vehicle Routing Problem. *Engineering Applications of Artificial Intelligence*, 23: 463-472.
24. Meer, K. (2007). Simulated Annealing versus Metropolis for a TSP Instance. *Information Processing Letters*, 104 (6): 216-219.
25. Moon, C., Kima, J., Choia, G., and Seob, Y., (2002). An Efficient Genetic Algorithm for the Traveling Salesman Problem with Precedence Constraints. *European Journal of Operational Research*, 140(3): 606-617.
26. Naderi, B., Tavakkoli-Moghaddam, R., and Khalili, M. (2010). Electromagnetism-Like Mechanism and Simulated Annealing Algorithms for Flowshop Scheduling Problems Minimizing The Total Weighted Tardiness and Makespan. *Knowledge-Based Systems*, 23(2):77-85.
27. Potvin, J. Y. (1996). Genetic Algorithms for the Traveling Salesman Problem. *Annals of Operations Research*, 63 (3): 337-370.
28. Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., and Wang, Q.X. (2007). Particle Swarm Optimization Based Algorithms for TSP and Generalized TSP. *Information Processing Letters*, 103: 169-176.
29. Tian, F. and Wang, L. (2000). Chaotic Simulated Annealing with Augmented Lagrange for Solving Combinatorial Optimization Problems. *Proceedings of the 26th Annual IECON*, 4: 2722-2725.
30. Tsai, C.F.. Tsai, C.W., and Tseng, C.C. (2004). A New Hybrid Heuristic Approach for Solving Large Traveling Salesman Problem. *Information Sciences*, 166: 67-81.
31. Yang, J., Wu, C., Lee, H.P., and Liang, Y. (2008). Solving Traveling Salesman Problem Using Generalized Chromosome Genetic Algorithm. *Progress in Natural Science*, 18: 887-892.
32. Yurtkuran, A. and Emel, E. (2010). A New Hybrid Electromagnetism-Like Algorithm for Capacitated Vehicle Routing Problems. *Expert Systems with Applications*, 37:3427-3433.